MULTI-SEMANTIC APPROACH TOWARDS A GENERIC FORMAL SOLVER OF TOOL PLACEMENT FOR PERCUTANEOUS SURGERY

Caroline Essert-Villard, Claire Baegert, Pascal Schreck
LSIIT - Université de Strasbourg, Boulevard Sebastien Brant, 67412 Illkirch, France
essert@unistra.fr, baegert@unistra.fr, schreck@unistra.fr

Keywords: Constraint solving, surgery planning, meta-programming

Abstract: In this paper, we study the multiple points of view when generalizing a method based on many criteria optimization, in the framework of percutaneous surgery planning. The aim of the prototype is to find an optimal position of surgical tools to help the surgeons in planning the intervention. We explain how, with a formal geometric solver and meta programming, we intend to build a modular tool, capable of being extended to new interventions, with few programming efforts.

1 INTRODUCTION

Nowadays, an increasing number of cancers are treated by using minimally invasive techniques, such as percutaneous radiofrequency, cryoablation, or microwave. These treatments have many advantages, but an important drawback is that the process generally involves a difficult and tedious planning phase, mainly relying on the study of the images of the patient (MRI, CT, ...), acquired before the intervention. Sometimes, a secure planning can not be found, prohibiting such an intervention.

In previous works, we presented a specific method to assist the surgeon in planning percutaneous RadioFrequency Ablations (RFA) for liver tumors, thanks to an automatic computation of an optimal needle trajectory. We observed a lot of similarities between the planning of this intervention and other percutaneous procedures (cryoablation), or kinds of surgery (*e.g.* Deep Brain Stimulation (DBS)).

This paper exposes how we analyzed similarities that leaded us to a generalization of our method, using a geometric formalization of the data and a formal geometric constraint solver. In the following section, after a brief review of other related works in the fields of surgery planning, we first recall our previous results. Then, we expose our analysis of the framework and explain our approach to extend the solver, and explain our choice of meta programming.

2 RELATED WORKS

In the domain of assistance to surgery planning, among which percutaneous interventions, various tools already exist. Most of them are simulators, modelling what will be the effect of a treatment, for a given placement of tools proposed by the surgeon. However, this forces the surgeon to perform himself/herself the trial and error search. Other authors proposed interesting attempts of automatic methods, but they have important drawbacks. In (Altrogge et al., 2006), authors do not take into account the presence of surrounding organs. Authors of (Lung et al., 2004) and (Adhami and Coste-Manière, 2003) confess a long computation time, and the first one algorithm is only in 2D. In (Brunenberg et al., 2007), authors restrict the research to a limited set of possible entry points, avoiding possibly good trajectories to be discovered. To our knowledge, very few automatic methods exist, and they are all very specific to one type of intervention. In this study, we focus on the genericity of the solving process for a set of percutaneous interventions sharing a lot of similarities.

In previous papers (Baegert et al., 2007a; Baegert et al., 2007b), we explained our method for providing automatically an optimal needle placement, in the framework of percutaneous RFA preoperative planning. The rules governing these interventions are collected from literature, observations, discussions with

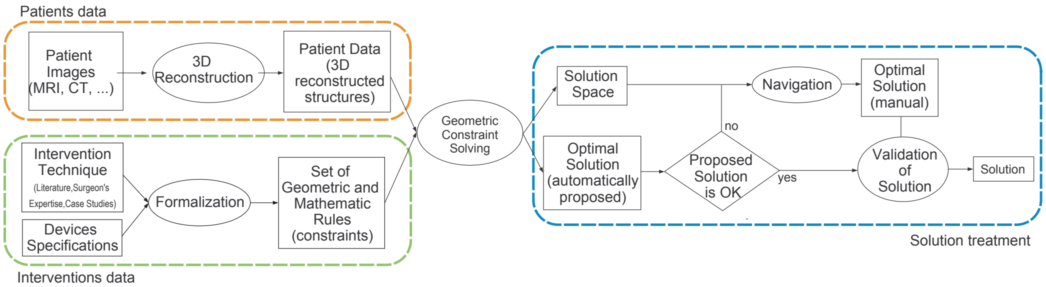
Patients data
Patient Images (MRI, CT, ...)
3D Reconstruction
Patient Data (3D reconstructed structures)
Intervention Technique
(Literature,Surgeon's Expertise,Case Studies)
Devices Specifications
Formalization
Set of Geometric and Mathematic Rules (constraints)
Interventions data
Geometric Constraint Solving
Solution Space
Navigation
Optimal Solution (manual)
no
Optimal Solution (automatically proposed)
Proposed Solution is OK
yes
Validation of Solution
Solution
Solution treatment

Figure 1: Process diagram

experimented surgeons. They are implemented in the program as functions, and solved using the patients images as input data. “Avoiding vital structures”, and “minimizing the damages on healthy cells” are examples of rules that are translated into “eliminating solutions crossing any organ mesh”, and “choosing those minimizing the volume of the effect of the treatment”. The output is a solution space constituted by the set of possible solutions, and an optimal solution among them. In Fig.1, a diagram showing the overall workflow is shown. The data specific to a type of intervention is defined once for all for a type of intervention, and is currently included in the code of the solver.

3 PERCUTANEOUS INTERVENTIONS

3.1 Motivation

Our previous method and the associated software were designed for a specific kind of intervention. The only modifiable parameters were the weighting coefficients applied to the different constraints. We aim at generalizing our approach, in order to be able to easily take into account: new constraints, new kinds of interventions, according to the different points of views of the actors of the process.

3.2 Different Points of View

In this field of application, we have to keep in mind that the final user as well as the main expertise contributor is a surgeon who is not necessarily a specialist in computer programming. So the description of the rules of the intervention, and the visualization, have to be adapted to the surgeon’s vocabulary and habits. Another point of view is the solver’s one: we have to provide our solver with a language and some data it is able to understand. A third point of view concerns the visualization. The computer needs useful understandable information for a proper display. Therefore, various points of view coexist, and it is necessary that the application deals with them.

3.3 Analysis of Knowledge

Many procedures involving the placement of one or more tools that can be assimilated to rays, with the aim of reaching and treating a target, have a lot of common points in their planning procedures. The two main similarities are the definitions of what constitutes a tool and what is a solution. We will illustrate our explanations with different examples: placement of needles for RFA and cryoablation, and DBS electrodes.

Definition A *tool* is defined by 3 features: its geometrical shape, the geometrical shape of the effect it produces, and the relative placement of the shape of the effect according to the shape of its support.

For instance the cryoablation’s tool is a needle, its effect is ellipsoidal and is centered at the location of the decompression chamber. All other information on the tool can be deduced from these 3 data.

The expert describes the tool with a specific terminology, for example “needle”, “electrode”, etc., and is able to make an approximative description of the effect. He can use either a geometric vocabulary like “cylindric shape, with a length of 5mm and a section of 3mm”, or a more fuzzy vocabulary such as “the effect looks like an olive”. The solver and the visualization module need a more concrete description of the tool and its effect, such as sets of voxels or meshes. Therefore all the information about the tool have to be collected, or deduced one from the other in any way.

Definition A *solution* is a placement of a tool. A placement is composed of a direction (2 degrees

Table 1: Examples of complex operators, with comprehensive names, and their profiles

lengthOfTool: tool → float	distToolOrgan: tool * solution * shape → float
volumeOfEffect: tool * solution → float	distEffectOrgan: tool * solution * shape → float
centerOfGravity: shape → point	coverZoneOrgan: tool * solution * shape → bool
dist2Pts: point * point → float	toolInsertionPoint: shape * solution → point

Table 2: Our current geometric universe

Types	Examples
point	insertion point, intersection organ/tool
shape	organ, zone of effect
tool	electrode, needle
solution	possible solution, optimal solution

of freedom (DOF)), an origin (3 DOF), and an extra value, for instance the intensity of the effect (1 DOF).

For RFA, a solution is a position of the tip of a needle, a direction of the main axis of the needle, and the power sent by the generator (size of the effect).

Here again, we have various points of view. The application (solver and visualization) works with 3D Cartesian coordinates, whereas the surgeon deals with various other position references: an insertion point on the skin, or another coordinate system such as Leksell coordinates in neurosurgery.

Another common point is that all the constraints we consider are based on the same constants: the images of the patient, that are segmented providing a set of organs (3D masks and meshes).

4 GEOMETRIC UNIVERSE

The meta programming approach we chose to use includes a geometric universe, and operators that can be combined to define the geometric constraints representing the rules of the surgical intervention.

4.1 Types

As a geometric universe, apart from usual types such as integers, floating numbers and booleans, we also use composed types, as we describe in Table 2. Those types are used for our constants and unknowns in the construction of our constraints.

Those composed types allow us to include for each entity all necessary information for the different points of view, and the different uses. For example, the "shape" type includes a 3D voxel mask (segmented from CT or MRI images) and a 3D mesh (reconstruction from a mask (organs), or by simple synthesis (effect)). Other properties such as the volume of a shape, can be deduced from these ones.

4.2 Operators

The differences between each surgical intervention are the values we give for the elements that define a tool, and the constraints to be solved. These constraints are expressed under the form of a combination of (predefined) operators, constants, and unknowns. We currently use very basic operators (such as plus, minus, multiply, and, or, etc.) as well as complex operators like the ones shown on Table 1. The terms that are formed using those operators and the constants can directly be written in the XML file describing the surgical intervention, that can be read by the interpreter.

5 APPLICATION TO RADIOFREQUENCY PLANNING

Our methodology for RFA planning explained in (Baegert et al., 2007a; Baegert et al., 2007b) was originally implemented in a specific solver. In order to validate our new approach, we transformed our original solver into a new generic solver, extracting the specific functions and replacing them by generic operators. The solver takes as an input trees of operators representing constraints, and performs a depth-first solving of the operators. We wrote the constraints using an XML syntax in a file dedicated to RFA interventions. An example of XML constraint is written in Table 3. In this example, the term constrains the trajectory (distance from insertion point to target) to be shorter than the length of the needle.

Finally, we experimented the solver to compare the solutions. The generic solver was tested on 10 patients cases we had already solved with the original version, and that were already validated by an expert (see (Baegert et al., 2007a) for detailed results). The resolution consists in finding the optimal position for the RFA needle, given a 3D scene representing the patient's anatomy, and given the constraints file for RFA interventions.

Table 3: Example of XML constraint, for Percutaneous RFA treatment

```
<strict_constraint name=”needle length restriction”>
    lower(dist2Pts(centerOfGravity(target), toolInsertionPoint(skin, solution)), lengthOfTool(tool))
</strict_constraint>
```

As expected, for each tested patient case, the results are identical in terms of precision of the solution (identical positions of tool and sizes of effect). The reason is that the whole computation operation used in the solver did not change, we only splitted the functions into small operators which combination described in the XML constraints file recreates the same computation scheme. The only difference in the results is in terms of computation time, that is a little bit slower with the generic version. We also expected this result, as an gain in genericity often comes with a performance loss. However, given that in both cases the total computation time of this planning process is performed in an average time of a few seconds (maximum experimental time for the worst case 2mn.), this increase in the computation time was considered as negligible and perfectly acceptable by the surgeons.

6 FUTURE REQUIREMENTS FOR MODULARITY

We presently dispose of a tested generic solver, directly able to find an optimal placement planning for RFA, but also for other percutaneous interventions with very similar tools and processes, if the appropriate constraint file is written. We are currently working on the constraint file for DBS in neurosurgery, with appropriate validation by experts.

For a more open genericity, we also need in future works to make sure of more extensions capabilities, in order to be able to include more surgical interventions types. We defined 3 categories of extensions, implying 5 different stages of various level of difficulty in the improvement of modularity of our method: new constraints using new operators, similar interventions using more than one needle (*e.g.* cryoablations), and interventions having other shapes of effect (*e.g.* radiotherapy).

7 CONCLUSION AND FUTURE WORKS

We described how we abstracted an existing solver of geometric constraints aiming at computing automatically an optimal placement of surgical tools for a specific intervention, to obtain a generic solver. We implemented a system loading a file describing the constraints of the surgical intervention for which a planning is required. The use of meta-programming allows us to describe the geometric constraints representing the rules of the surgical intervention with a language more accessible than a programming language, and with a geometric universe and operators that could be redefined on the fly in the future.

Further works remain to be done in order to be even more generic and to extend to more surgical interventions. This study showed us that it will be feasible in a reasonable time, and with a reasonable amount of work. Besides the future extensions of the solver, we will also have to write the constraints of the other aimed surgical interventions, and validate them with experts.

REFERENCES

Adhami, L. and Coste-Manière, E. (2003). Optimal planning for minimally invasive surgical robots. *IEEE Transactions on Robotics and Automation : Special Issue on Medical Robotics*, 19(5):854–863.

Altrogge, I., Kröger, T., Preusser, T., Büskens, C., Pereira, P., Schmidt, D., Weihusen, A., and Peitgen, H. (2006). Towards optimization of probe placement for radiofrequency ablation. In *MICCAI'2006*, volume 4190 of *LNCS*, pages 486–493.

Baegert, C., Villard, C., Schreck, P., and Soler, L. (2007a). Multi-criteria trajectory planning for hepatic radiofrequency ablation. In *MICCAI'2007*, volume 4791 of *LNCS*, pages 584–592.

Baegert, C., Villard, C., Schreck, P., and Soler, L. (2007b). Trajectory optimization for the planning of percutaneous radiofrequeny ablation on hepatic tumors. *Computer Aided Surgery*.

Brunenberg, E., Vilanova, A., Visser-Vandewalle, V., Temel, Y., Ackermans, L., Platel, B., and ter Haar Romeny, B. (2007). Automatic trajectory planning for deep brain stimulation: A feasibility study. In *MICCAI'2007*, volume 4791 of *LNCS*, pages 584–592.

Lung, D., Stahovich, T., and Rabin, Y. (2004). Computerized planning for multiprobe cryosurgery using a force-field analogy. *Computer Methods in Biomechanics and Biomedical Engineering*, 7(2):101–110.